INTERPRETING MULTILINEAR REPRESENTATIONS IN SPEECH

Julie Carson-Berndsen and Michael Walsh
University College Dublin, Ireland

ABSTRACT: This paper discusses the interpretation of multilinear representations of speech utterances using a computational linguistic model. The model uses a feature-based finite state automaton representation of phonotactic constraints and axioms of event logic to provide a multilinear representation with a temporal interpretation. The asynchronous nature of the features in multilinear representations allows coarticulation to be modeled and the phonotactic automaton representation of permissible sound combinations in a language allows not only actual but also potential (well-formed) syllable structures to be recognised. The paper illustrates how a computational linguistic model can provide a robust solution to the problems of coarticulation and out-of-vocabulary items in speech recognition.

1. INTRODUCTION

In this paper we present a framework which uses finite state automaton representations of phonotactic constraints in the syllable domain and axioms of event logic to interpret multilinear representations of speech utterances. The framework builds on the model of *Time Map Phonology* (Carson-Berndsen, 1998) but differs in that phonotactic constraints are not restricted to one particular feature classification and that other phonotactic descriptions for other languages and based on other feature sets can be defined and evaluated. In what follows, we introduce the framework by adhering to one simple example which illustrates the attributes of the approach. Section 2 introduces the notion of a multilinear representation and briefly discusses the computational linguistic model which is used for interpretation. Section 3 discusses the generic framework for the computational linguistic model and the principal components of this framework are illustrated with respect to an example utterance in sections 4 and 5. Section 6 describes one method of constraint relaxation to cater for underspecification in the input representations and section 7 concludes with some discussion of future directions.

2. REPRESENTING SPEECH UTTERANCES

In this paper we focus on a particular representation of speech utterances based on phonological features. Phonological theory distinguishes between segmental and non-segmental approaches to the description of spoken representations. The segmental approach assumes that the spoken language representation can be sliced up into non-overlapping segments in which features are synchronous. The non-segmental approach as found in autosegmental phonology (see Goldsmith, 1990) prefers to treat spoken language representations in terms of tiers of autonomous features (autosegments) which can spread across a number of sounds. This idea has been developed further in computational phonology by Bird (1995) and Carson-Berndsen (1998) among others. The advantage of this type of representation is that coarticulation can be modeled by allowing features to overlap. While speech recognition has traditionally attempted to segment speech utterances into non-overlapping segments in terms of phones, phonemes or demisyllables etc. (but cf. Deng & Wu, 1996), our approach assumes an autonomous treatment of features in line with autosegmental phonology resulting in a multilinear representation of features which is interpreted by an event-based computational linguistic model of phonology. The representation is not unrelated to the type of structures seen in the work of Browman & Goldstein (1989) in their model for articulatory synthesis.

The computational linguistic model of phonology is the *Time Map* model (cf. Carson-Berndsen, 1998, 2000) which uses phonotactic automata (network representations of the permissible combinations of sounds in a language) and axioms of event logic to interpret multilinear representations. The *Time Map* model distinguishes between two time domains: the *absolute* (or signal) *time* domain in which features are treated as events with temporal endpoints and the *relative time* domain in which only the temporal relations (overlap and precedence) between the events are relevant. Although the multilinear representation, which acts as input to the model, is in the absolute time domain, parsing is carried out exclusively in the relative time domain using only the relations. Parsing is guided by the phonotactic automaton which specifies top-down constraints on the overlap and precedence relations which can occur in a particular language.

The type of representation expected as input to the *Time Map* model can be seen in figure 1 which depicts the neologism "*stramsam*" (a potential syllable followed by an actual or lexicalised one) in terms of a multilinear representation of autonomous features. The advantage of this representation is that the features are not synchronous and therefore a strict segmentation into phone-like units is not required.

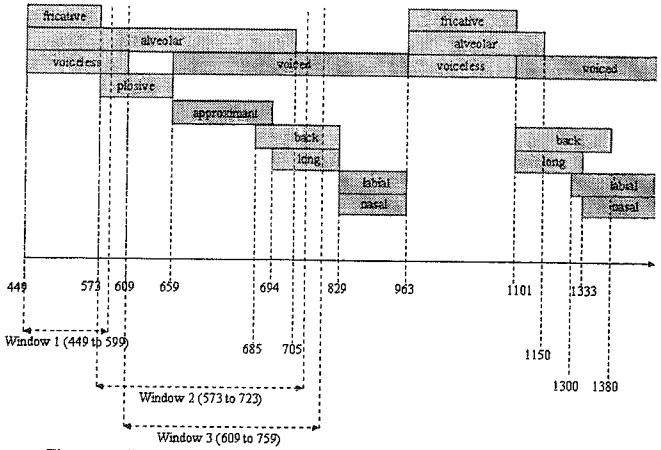

Figure 1. The neologism "*stramsam*" in terms of its feature and temporal structure. The time-line is not scaled.

Of course, the use of linguistic features in speech recognition is not new. Just recently the role of linguistic features in speech recognition has received more attention (cf. Ostendorf (2000), Koreman et al. (to appear), Salomon & Espy-Wilson (1999)). More specifically, King et al. (2000) describe a phonological feature-based system which uses a neural net to produce a feature representation not dissimilar to the multilinear representations discussed above. Our work does not contradict any of these approaches since we are concerned not with feature extraction but with interpretation at the phonological level. Our system does require, however, that features can be treated autonomously, as those in the King et al. (2000) model would seem to allow. As will be seen in section 4 the actual choice of features to be used in the model is freely definable. In this paper we use an IPA-like feature classification throughout.

In the rest of this paper we discuss the workings of the system by use of the single example of figure 1 which has been recorded, manually labeled and then manipulated to ensure that the features are not synchronous. This has been done purely to illustrate that the approach can deal with asynchronous feature representations as is required to model coarticulation phenomena (overlap of properties); it would be more difficult to present the details of the approach with a larger utterance and using a complete phonotactics of the language. The approach has, however, been implemented and evaluated within a speech recognition system using a complete phonotactics of German (cf. Carson-Berndsen, 1998 for details). Note that in the representation in figure 1 each sound[1] is assumed to be fully specified. This is of course not always the case. We will return to this issue in section 6.

3. LIPS: A generic framework for the *TIME MAP* MODEL

Given the multilinear representation introduced in the previous section the next question is to determine how these features, which are generally asynchronous (leading to overlap), can be parsed. The *Time Map* model provides us with a robust non-segmental phonology for modelling coarticulation and out-of-vocabulary items. Essentially the *Time Map* model takes feature extractor output and maps the occurrence of features from absolute (signal) time to relative time, i.e. it determines how features present in a speech signal temporally relate to one another. It achieves this by using an event logic, a set of axioms which when applied to the temporal properties of two features (e.g. when they start and

end) determines whether or not they overlap or precede each other in time. The Language Independent Phonotactic System (LIPS) is the generic framework for the *Time Map* model and successfully encompasses the axioms of the event logic, thus facilitating the mapping from actual speech signal time (absolute time) to the temporal relations between features. In addition LIPS also incorporates a finite state methodology which enables users to create their own phonotactic automata for individual languages by means of a graphical user interface.

A phonotactic automaton is one which models the legal combinations of sounds in a given language. Thus for English, a phonotactic automaton representing English syllable structure will allow an [s] followed by a [p] in syllable-initial position, but not the other way around. The language independence functionality and generic characteristics of the system stem from two significant attributes of the interface. Firstly the interface permits users to easily construct a phonotactic model of their desired language by allowing them to specify the structure (nodes) and contents (arcs) of a finite state syllable automaton. Secondly, while providing an IPA-like feature set from which they can define the contents of their automaton, there is also an option to add new features. Thus LIPS is neither dependent on any language nor on any feature set. Furthermore, this ability to employ any feature set facilitates the investigation of the type of linguistic features which may be most relevant to speech recognition applications and could therefore provide important insights for speech recognition models currently proposing the use of linguistic features (see section 2). The system comprises two principal components, the network generator and the parser, which are outlined in the following sections. In what follows we have restricted the description to a single multi-valued feature set, but in principle any feature system may be substituted as a long as it is defined in the network generator.

4. THE NETWORK GENERATOR

The network generator interface allows the user to enter node values and to select from a list of feature overlap relations those that a given arc is to represent (see figure 2). Alternatively users can specify their own features. Characteristics of the *Time Map* model are incorporated implicitly in that features precede other features in the network as arcs precede other arcs, i.e. if according to the phonotactics of the language in question an [s] precedes a [p] in syllable-final position, then the user can define an automaton with the features (generally underspecified) representing the [s] on an arc which immediately precedes the arc containing the features representing the [p], at the syllable-final position in the network.

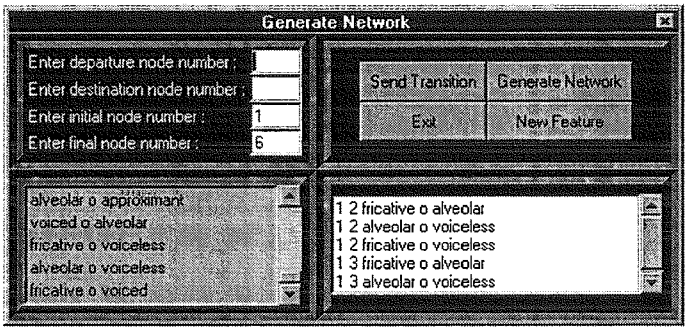

Figure 2. The Language Independent Phonotactic System's interface for generating phonotactic automata.

It is important to note that the network generator constructs feature-based networks and feature overlap relations are selected explicitly. Thus a transition might look like the following:

<1 2 alveolar ° voiceless>

which states that the arc between nodes 1 and 2 specifies the temporal constraint that an alveolar feature must overlap a voiceless feature. When the user has completed the network specification the system generates a list of transitions representing the automaton in Figure 3.

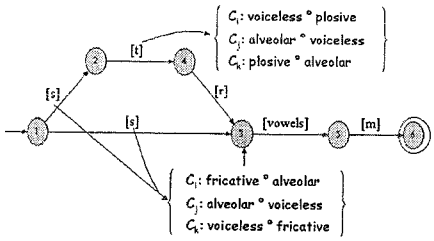

Figure 3. Phonotactic automaton representing a very small subsection of the phonotactics of English.

The automaton specifies top-down constraints on the overlap and precedence relations which occur in particular phonotactic positions within the syllable. This information is used by the parser which is the topic of the next section.

5. THE PARSER

LIPS employs a top-down parsing strategy and a breadth-first search strategy to ensure that all possible paths through the network are attempted. Rather than mapping all absolute time information into relative time, the parser only scans for those temporal overlaps which are anticipated by the network. In this way the search space and processing time are minimised. Parsing is best explained through exemplification.

We return to the example network generated above as depicted in figure 3; of course this represents only a very small subsection of the phonotactics of English. The finite-state automaton will recognise such syllables as *strum, stram, sam, sim, sum, am, im,* etc., some of which are actual syllables (and even monosyllabic words) of English, and all of which are phonotactically well-formed. In general the phonotactic automaton describes all the phonotactic possibilities of the language (actual and potential syllables). In our example the feature extractor output for the utterance "*stramsam*", as illustrated graphically in figure 1, serves as the input to the parser. Feature events in the input might look like the following:

449 s fricative
449 s alveolar
449 s voiceless
573 e fricative
609 e voiceless

indicating that a fricative feature and a voiceless feature and an alveolar feature begin at 449ms; the fricative feature ends at 573ms and the voiceless feature at 609ms. Feature overlaps which are anticipated by the network are scanned for within a parametrisable millisecond window, currently set to 150ms. Therefore at the beginning, according to the phonotactic automaton in figure 3, either an [s] or a vowel is expected by the network, i.e. the automaton starts looking from node 1 and node 3. Figure 4 represents the interaction between the phonotactic automaton of figure 3 and the temporal / feature structure of the "*stramsam*" recording of figure 1. It serves as a useful illustrative guide through the parsing process.

Starting at the temporal value of the beginning of the first event, in this case 449ms, a window of 150ms is imposed ("Window 1 (449 to 599)" in figure 4) which allows for incremental processing. Clearly the features which represent the [s] (fricative, alveolar, and voiceless) are found within the window, but not those of a vowel. The temporal values of these features, and only these features, are then examined, according to the axioms of the event logic, to see if the features overlap. Given that in this case the features clearly do overlap, thus satisfying the constraints for the [s] as specified in figure 3, the target nodes of the transitions from node 1, namely nodes 2 and 3, are then predicted as the next nodes in the automaton from which to search. The smallest, average or largest endpoint (parametrisable) of the features found serves as the start-point for the window associated with these nodes. In our example the smallest endpoint is assumed. Hence the second window starts at 573ms ("Window 2 (573 to 723)").

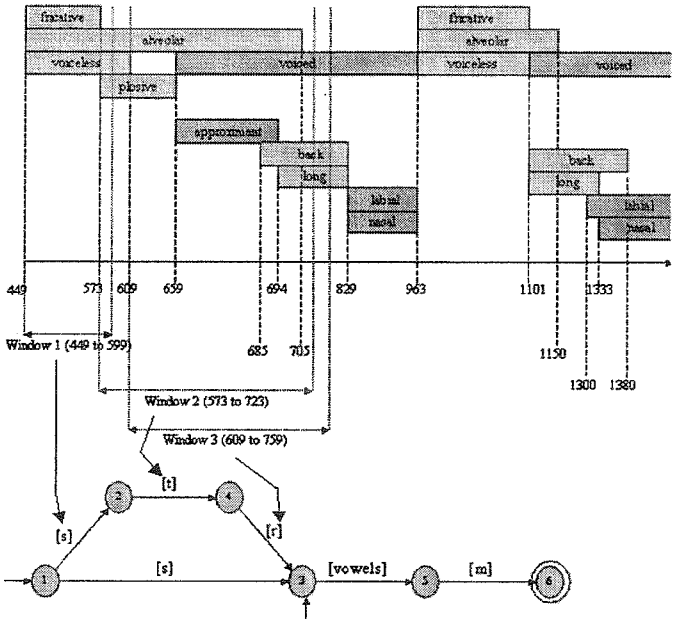

Figure 4. Interaction between the input and the automaton.

Prior to searching from nodes 2 and 3, the successors of the first level of search, the parser carries out an immediate precedence check. This entails determining if, for all elements of the current level of search, an anticipated feature is present within a parametrisable "sub-window", in this case set to 30ms. Node 3, starting at 573ms, is anticipating vowel features, such as voicing. However none of these are present between 573ms and 603ms. On the other hand node 2, also starting at 573ms, is anticipating features for a [t], such as a plosive, which is indeed present. This results in node 3 being excluded from the search space and node 1 being retained, as the check indicates that features representing the [t] precede those that represent a vowel. The parser then continues searching within the second window for the features on the arc between nodes 2 and 4, i.e. (voiceless, plosive, alveolar) those that correspond to [t]. These features are present in the window and overlap resulting in the target node, node 4, as the only successor for this level of search. Once again the smallest endpoint of the features found, in this case 609ms (the endpoint of the voiceless feature), is used as the start-point for the next window ("Window 3 (609 to 759)").

This process continues, with precedence checks taking place with each level of search, until the end of the syllable is reached (node 6) at which point nodes 1 and 3 are predicted for the next syllable i.e. the automaton starts again, searching from nodes 1 and 3, with a new temporal start-point. Parsing ends when there are no more alternative paths to take. All structures which have paths from node 1 to node 6 are output.

6. RESOLVING POTENTIAL PROBLEMS IN THE INPUT

The graphical representation of the utterance "*stramsam*" as illustrated in figure 1 shows clearly that although the features are often asynchronous, all the feature overlap relations anticipated for a given sound at a given stage in the network are present. In continuous speech this is not necessarily the case of course. For example, the recognition of some sound [ω] might be dependent on features *x, y* and *z* all overlapping. However although *x* might overlap *y*, and *z* might overlap *y*, *x* might not overlap *z*. Thus only two of the three overlap relation constraints would be satisfied (i.e. constraint relaxation is applied). LIPS caters for this underspecification in the input by allowing users to rank each feature

overlap constraint in a transition and to attribute a threshold value to the transition as a whole. Thus for a transition from node 1 to node 2 the following constraints might apply:

$$Ci \rightarrow x \circ y \quad Rank = 5$$

$$Cj \rightarrow z \circ y \quad Rank = 3$$

$$Ck \rightarrow x \circ z \quad Rank = 4$$

According to these constraint rankings and a transition threshold of 7, if any two of the overlap relations were present in the input then the transition would be successfully parsed (i.e. the rank values of the overlaps found would be summed and compared against the threshold). The example above is arbitrary and is employed for exemplification. The user should rank constraints in a motivated fashion, that is they should be ranked according to results of statistical analysis on the data in question or on knowledge of the particular feature set employed.

7. CONCLUSION

This paper has discussed an approach to interpreting multilinear representations of speech utterances using a computational linguistic model of phonology. The aim of the paper has been to describe how the approach deals with asynchronous features and therefore a simple example utterance was chosen for illustration purposes. In general, the model assumes that a complete phonotactics of a language is defined via the network generator. While our current model caters for German and English phonotactics, future work involves evaluating different feature sets and languages using the LIPS framework.

REFERENCES

Bird S. (1995): Computational Phonology: A constraint-based approach. (Cambridge University Press: Cambridge).

Browman, C & L. Goldstein (1989): "Articulatory gestures as phonological units". In: *Phonology 6*, 201-251.

Carson-Berndsen J. (1998): Finite State Models and Event Logics in Speech Recognition, (Kluwer Academic Publishers:Dordrecht).

Carson-Berndsen J. (2000) "Finite State Models, Event Logics and Statistics in Speech Recognition", Philosophical Transactions of the Royal Society, Series A, Volume 358, issue no. 1769, 1255-1266.

Deng L & J. J.-X. Wu (1996): "Hierarchical partition of the articulatory state space for overlapping feature based speech recognition". Proceedings of ICSLP '96, vol 4. , 2266-2269.

Goldsmith, J. (1990): Autosegmental and Metrical Phonology. (Basil Blackwell: Cambridge MA).

King S.; P. Taylor; J. Frankel & K. Richmond (2000): "Speech Recognition via Phonetically-Featured Syllables". In: Phonus, Institute of Phonetics Saarbrücken.

Koreman J; B. Andreeva & W. Barry (to appear): "Phonetic features in ASR: A linguistic solution to acoustic variation". Proceedings of the 7th Conference on Laboratory Phonology, Nijmegen.

Ostendorf M. (2000): "Incorporating linguistic theories of pronunciation into speech recognition models", *Philosophical Transactions of the Royal Society*, Series A, Volume 358, issue no. 1769.

Salomon A. & C. Espy-Wilson (1999): "Automatic Detection of Manner Events based on Temporal Parameters". In: Proceedings of Eurospeech 99, vol 6, 2797-2800.

---

[1] Note that we refrain from using the terms *phone* and *phoneme* here as our system is purely feature based. The phone symbols are mnemonic for sets of feature overlap constraints.